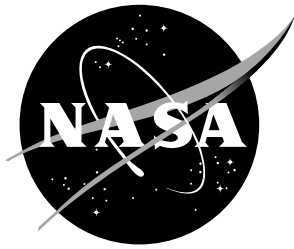


NASA/TM-2016-219216



# Using 3-D Numerical Weather Data in Piloted Simulations

*Taumi S. Daniels*  
*Langley Research Center, Hampton, Virginia*

---

July 2016

## NASA STI Program...in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collection of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

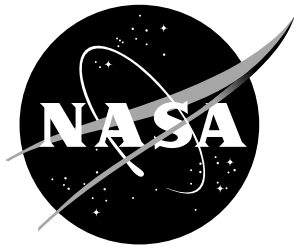
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI Program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question to [help@sti.nasa.gov](mailto:help@sti.nasa.gov)
- Phone the NASA STI Information Desk at 757-864-9658
- Write to:  
NASA STI Information Desk  
Mail Stop 148  
NASA Langley Research Center  
Hampton, VA 23681-2199

NASA/TM-2016-219216



# Using 3-D Numerical Weather Data in Piloted Simulations

*Taumi S. Daniels*  
*Langley Research Center, Hampton, Virginia*

National Aeronautics and  
Space Administration

Langley Research Center  
Hampton, Virginia 23681-2199

---

July 2016

## Acknowledgments

The author would like to thank Dr. Steve Young, Emory Evans, and Philip Schaffner, and Dr. Andrew Moore. This work was funded by the NASA Airspace Operations and Safety Program.

<p>The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.</p>
---

Available from:

NASA STI Program / Mail Stop 148  
NASA Langley Research Center  
Hampton, VA 23681-2199  
Fax: 757-864-6500

## Abstract

This report describes the process of acquiring and using 3-D numerical model weather data sets in NASA Langley's Research Flight Deck (RFD). A set of software tools implement the process and can be used for other purposes as well. Given time and location information of a weather phenomenon of interest, the user can download associated numerical weather model data. These data are created by the National Oceanic and Atmospheric Administration (NOAA) High Resolution Rapid Refresh (HRRR) model, and are then processed using a set of Mathworks' Matlab<sup>TM</sup> scripts to create the usable 3-D weather data sets. Each data set includes radar reflectivity, water vapor, component winds, temperature, supercooled liquid water, turbulence, pressure, altitude, land elevation, relative humidity, and water phases. An open-source data processing program, `wgrib2`, is available from NOAA online, and is used along with Matlab scripts. These scripts are described with sufficient detail to make future modifications. These software tools have been used to generate 3-D weather data for various RFD experiments.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Overview of Process</b>	<b>4</b>
<b>3</b>	<b>Acquiring Weather Data</b>	<b>5</b>
<b>4</b>	<b>Software Requirements</b>	<b>7</b>
4.1	Source Code wgrib2 . . . . .	7
<b>5</b>	<b>User Inputs</b>	<b>7</b>
<b>6</b>	<b>Extract Weather Data Fields</b>	<b>8</b>
<b>7</b>	<b>Modifications to Weather Data Fields</b>	<b>9</b>
7.1	Altitude Modification . . . . .	9
7.2	Vertical Wind Modification . . . . .	10
7.3	Horizontal Wind Modification . . . . .	10
<b>8</b>	<b>Derived Weather Fields</b>	<b>10</b>
8.1	Radar Reflectivity . . . . .	10
8.2	Turbulence Parameter . . . . .	10
8.3	Aircraft Icing . . . . .	11
<b>9</b>	<b>Script Summary</b>	<b>11</b>
<b>10</b>	<b>Conclusions</b>	<b>11</b>
<b>A</b>	<b>runhrrr.m listing</b>	<b>14</b>
<b>B</b>	<b>extract.m listing</b>	<b>16</b>
B.1	Subroutine reversez.m listing . . . . .	19
<b>C</b>	<b>hgtcorr.m listing</b>	<b>20</b>
<b>D</b>	<b>vvelcorr.m listing</b>	<b>21</b>
<b>E</b>	<b>windsrot2d.m listing</b>	<b>23</b>
<b>F</b>	<b>genrrf.m listing</b>	<b>26</b>
<b>G</b>	<b>tke.m listing</b>	<b>30</b>
<b>H</b>	<b>genslw.m listing</b>	<b>32</b>

## List of Acronyms, Symbols, and Abbreviations

AGL	=	Above Ground Level
CLWMR	=	Cloud Water Mixing Ratio
CICE	=	Cloud Ice Mixing Ratio
CMF/RFD	=	Cockpit Motion Facility's Research Flight Deck
CONUS	=	Continental United States
DPT	=	Dew Point Temperature
FAA	=	Federal Aviation Administration
GRIB	=	Gridded Binary
GRLE	=	Grauple Mixing Ratio
HGT	=	Geopotential Height
HRRR	=	High Resolution Rapid Refresh
KMEM	=	Memphis International Airport
MET	=	Meteorological
MSL	=	Mean Sea Level
NetCDF	=	Network Common Data Form
NM	=	Nautical Mile
NOAA	=	National Oceanic and Atmospheric Administration
RFD	=	Research Flight Deck
RH	=	Relative Humidity
RRF	=	Radar Reflectivity
RUC	=	Rapid Update Cycle
RWMR	=	Rain Water Mixing Ratio
SLW	=	Super-cooled Liquid Water
SNMR	=	Snow Mixing Ratio
SPFH	=	Specific Humidity
TKE	=	Turbulent Parameter
TMP	=	Temperature
UGRD	=	N-S Component of Wind
UR	=	Corrected U-component of Wind
UTC	=	Coordinated Universal Time
VGRD	=	E-W Component of Wind
VR	=	Corrected V-component of Wind
VVEL	=	Vertical Velocity (Pressure)
VVMS	=	Vertical Velocity
WDIR	=	Wind Direction
WMAG	=	Wind Magnitude
Z	=	Geometric Height

# 1 Introduction

There is a need for 3-D weather or other reproducible weather data for flight simulation experiments conducted in NASA’s Cockpit Motion Facility Research Flight Deck (CMF/RFD), or for other reproducible weather simulations. This is especially true for experiments that require reproducibility across repeated runs or simulations within the same experiment. A realistic weather data set should contain all of the relevant data fields for a full immersion experience during the experiment. In a simulation environment such as the RFD, weather data can be used as part of any experiment.

The software tools described here are reported in [1] and [2]. This report serves to document the software tools and provide the necessary information to make use of other similar weather data sets.

NOAA runs the High Resolution Rapid Refresh (HRRR) numerical weather model that ingests weather observations from a variety of instruments, and creates numerous forecast products [3, 4]. These products, or state variable weather fields, consist of temperature, water vapor, pressure, winds, and water phases. All of the products are generated every hour, including the hourly forecasts, or 1 hour, 2 hour, etc., up to 15 hour forecasts. One note of distinction is the analysis output, or the HRRR output at 0 hour. This is essentially an HRRR model run using all of the most recent data (including results from past runs). This 0 hour run populates the output grid with solutions to the state equations. A particular NOAA grid is used that encompasses the Continental United States (CONUS) and includes about 50 altitude or pressure levels.

The following sections describe steps for converting NOAA HRRR weather data into unique and separate data fields. Each step of the process is outlined along with descriptions of the associated Matlab scripts. Sample figures of weather data are also provided. Matlab scripts are listed in the appendices.

# 2 Overview of Process

Creation of a weather data set consists of a series of steps. This process, as depicted in Figure 1, creates a set of weather data fields. In this report, a field is a variable in 3-D, with dimensions latitude, longitude, and altitude. The process starts by selecting a significant weather event, such as a meso-scale convective complex. Ideally, this event occurs over the CONUS region of interest. For an RFD simulation, a particular airport often defines the region of interest. As the range setting of an on-board weather radar may be up to 300 nm, this region of interest must be expanded beyond the airport to include geographic points that are at least 300 nm distance from all possible aircraft locations. In addition, the user must select the UTC time at which the significant weather is occurring.

This time and location information is then used to select the NOAA HRRR files to download. Once acquired, the HRRR file(s), in GRIB2 format, are processed by a series of Matlab scripts, each creating or adjusting the desired output weather data fields.



### 3 Acquiring Weather Data

NOAA maintains an online archive of hourly analysis and forecast products for the previous day up to the current time for each hourly run of the HRRR. The website can be accessed via anonymous FTP via the URL: `ftp://ftp.ncep.noaa.gov/pub/data/nccf/nonoperational/com/hrrr/prod/`.

Accessing this website via a web browser will present a webpage with two folder links. Each link is a dynamically changing folder with a name that corresponds to the current date and previous date. The two date folder names have "hrrr." prepended to the date string. Located within the previous date folder are various GRIB2 files and other information files for the 24 hourly HRRR runs and all the forecasts for each hour. The current day folder includes all those files from midnight (UTC) up to the current hour. The only files of interest are those that have ".grib2" as the extension.

For example, a sample directory is `hrrr.20160405`. This directory (and link) example is for HRRR runs generated on April 5, 2016. Within that folder resides a set of GRIB2 and other files. The set is complete for the prior day's folder, and incomplete for the current day depending on when it is accessed.

An example file name is `hrrr.t09z.wrfprsf00.grib2`. This analysis file includes "wrfprsf00" as part of the file name. These are the pressure level based forecast products generated at the beginning of HRRR runs. The UTC time for this example run is the "t09z" part of the file for weather events that occurred between 8:00 UTC to 9:00 UTC. NOAA collects weather observations up till the top of each hour, then commences each HRRR run.

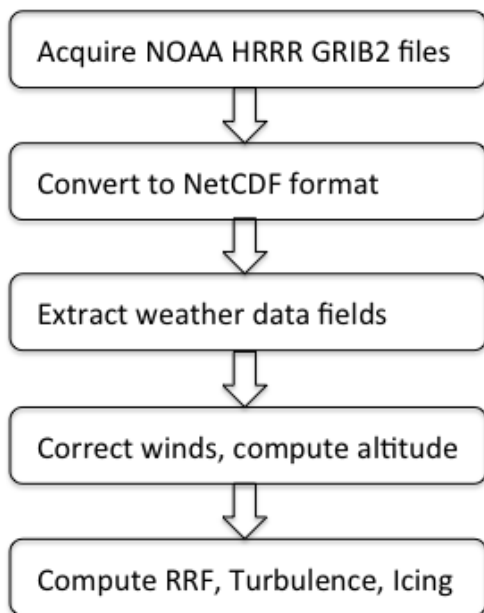


Figure 1. Process Overview.

The following website [http://radar.weather.gov/ridge/Conus/full\\_loop.php](http://radar.weather.gov/ridge/Conus/full_loop.php) is a National Weather Service Enhanced Radar Image, that loops over the previous hour. This imagery can be used to select the time when suitable weather is impacting the CONUS region of interest.

The Lambert Conformal Projection is used by NOAA for the HRRR data grid. Parameters for this projection are: central Latitude 38.5N, central Longitude 262.5W, Pole Latitude 90N, Pole Longitude 0W.

HRRR analysis and forecast files are provided for the CONUS domain, as depicted in Figure 2. The domain includes 1800 x 1060 grid nodes over the region with corners (SW-NW-NE-SE) at 21.13812N - 237.2805W, 47.84364N - 225.9014W, 47.84364N - 299.09863W, and 21.13812N - 287.71954W, and 50 pressure (altitude) levels. Each grid node has approximately 3 km horizontal spacing, Finer spatial resolution required by a simulated on-board aircraft radar display must be obtained by interpolation. The Matlab scripts described here compute radar reflectivity fields with 3 km spacing. In order to create an on-board aircraft radar image of convection, radar reflectivity must be spatially interpolated in 3-D into radar plan position indicator (PPI) grid.

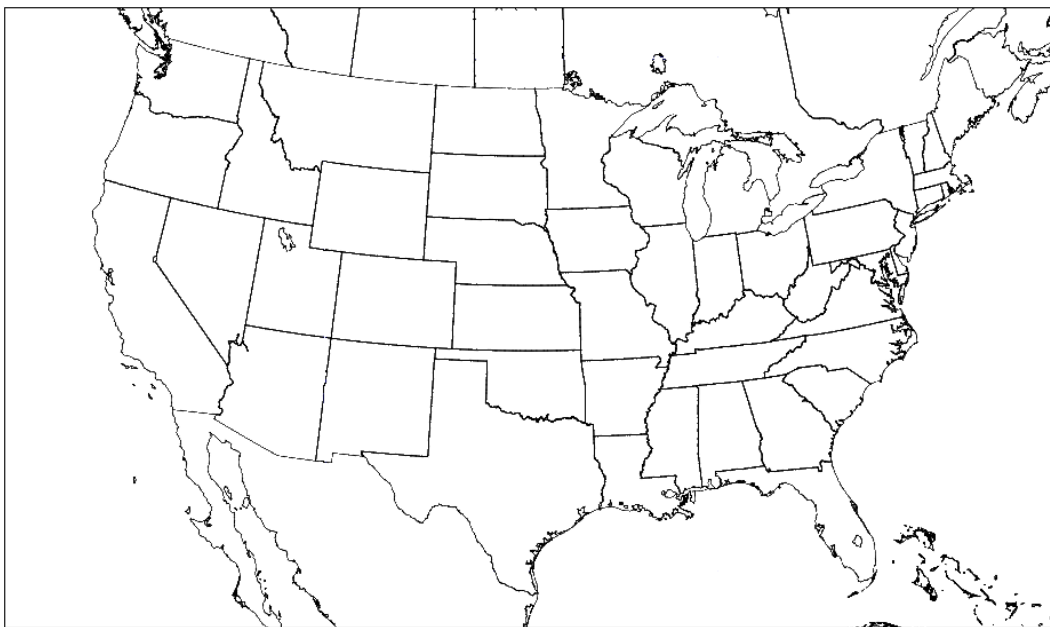


Figure 2. HRRR Domain.

In addition, other files and folders reside in the two HRRR folders. These unused products include: forecast files named “wrfprs01”, “wrfprs02”, etc.; files named “wrfnat” defined at different altitude levels, mainly near the Earth; files with the extension “.idx”; and subfolders named “bufrsnd”.

If the user requires 4-D weather data, 3-D data at multiple time instances, then two or more GRIB2 files that are contiguous in time can be processed to create the weather data fields. Weather data fields in 3-D can be interpolated in time to provide the added dimension.

## 4 Software Requirements

To process the GRIB2 files, the user must create the `wgrib2` executable, make it available to the user account processing the data, and make it accessible from the Matlab graphical user interface environment. All of the Matlab scripts listed in the Appendices were written for the Apple Mac under OS-X 10. In order to use the scripts on a Microsoft Windows based system, the directory paths need to be redefined by converting each forward slash to a backward slash. The location of the `wgrib2` executable must be made accessible via the appropriate system environment variable.

### 4.1 Source Code `wgrib2`

At the website <http://www.cpc.ncep.noaa.gov/products/wesley/wgrib2/>, NOAA hosts the source code, compile and usage instructions, and machine specific information needed to create the `wgrib2` executable. This software tool performs many of the pre-processing operations needed to convert GRIB2 formatted HRRR data into weather data fields.

Table 1. HRRR Weather Data Fields

Field Name	Description	Units
HGT	Geopotential Height	m
TMP	Temperature	K
DPT	Dew Point Temperature	K
RH	Relative Humidity	%
SPFH	Specific Humidity	kg/kg
VVEL	Vertical Velocity (Pressure)	Pa/s
UGRD	N-S Component of Wind	m/s
VGRD	E-W Component of Wind	m/s
CLWMR	Cloud Water Mixing Ratio	kg/kg
CICE	Cloud Ice	kg/m <sup>2</sup>
RWMR	Rain Water Mixing Ratio	kg/kg
SNMR	Snow Mixing Ratio	kg/kg
GRLE	Grauple Mixing Ratio	kg/kg

## 5 User Inputs

Processing files implemented in Mathworks' Matlab<sup>TM</sup> are listed in the Appendices. All processing is controlled by `runhrrr.m`, which calls other scripts. In this file, user settings are made in lines 24 through 29. Input and output directories are specified using the `gribdir` and `hrrrdir` string variables, respectively. Also, the region of interest latitude and longitude are defined using `targetlat` and `targetlong` string variables.

## 6 Extract Weather Data Fields

In Appendix B, a listing of the Matlab script `extract.m` is given. This code produces each of the weather data fields listed in Table 1. If needed, weather data fields `RH`, `CICE`, `SNMR`, and `DPT` can be made available by editing line 6 in `extract.m`. All of the state variable weather fields are provided for 50 pressure levels from 1013.2 mb to 50 mb, or from the surface to about 21,000 m (70,000 ft).

The `extract.m` script begins by establishing locations of GRIB2 files, output directory name, and the geographical location of the target region. In this example, the Memphis International Airport (KMEM) at Memphis, TN is located approximately at 35N 270W. An approximate 600 km per side roughly square region with Memphis at the center has coordinates ranging from 32° to 38° in Latitude and 266° to 274° in Longitude. These values are input to the `wgrib2` executable.

In the `extract.m` script, lines 64-147, a conditional section of code determines if the file `memetax.mat` exists in the current directory, and if not then creates it. To do this, three calls to `wgrib2` are performed. The first call selects a `HGT` GRIB2 file. The second call subselects the region of interest using the defined Latitude and Longitude variables. Finally, the third call converts the output to NetCDF format. From this NetCDF file, the script reads and creates the variables `latitude`, `longitude`, and surface (above sea level), `msl_m`, as 2-D matrices. Matlab does not recognize the GRIB2 format but does have built-in commands to manipulate NetCDF files.

An example of the selected domain outlined by `latitude` and `longitude` is illustrated in Figure 3. The shape of this region as determined by the Lambert projection as shown in the figure.

The `extract.m` script continues by creating a fourth output data field, pressure levels. This one dimensional array is extracted from the same `HGT` GRIB2 file, with four calls to `wgrib2`. Again, the first call extracts the `HGT` variable, the second subselects the first 36 of 50 pressure levels, the third subselects the region of interest, and the fourth converts to NetCDF format. The resulting pressure level array, `plevel`, is one-dimensional. All temporary files are deleted.

The Lambert projection applies to each output data grid. Subsequent Matlab scripts perform additional processing, including modification of the wind direction for this Lambert projection, as described in Section 7.3. Aeronautical charts also use the Lambert projection, though at different reference parallels. No conversion from this NOAA Lambert projection to any other, such as the Lambert projection used in FAA Aeronautical Charts, is performed.

In the `extract.m` script, lines 151-221, each iteration of the FOR loop, processes a different GRIB2 file from the specified input directory. Each file is processed by the FOR loop, lines 162-219, to generate each of the weather data fields in separate files. All generated output files have the same name as the input GRIB2 file with the appropriate weather variable appended at the end.

The weather data files are generated using four calls to `wgrib2`. The first call selects from the input GRIB2 file the weather variable field of interest into a temporary GRIB2 file. This temporary file is then used in the second call to down select the 36 of 50 pressure levels of interest. The output is a temporary file that is used in the third call to down select the region of interest. Finally, the fourth call converts

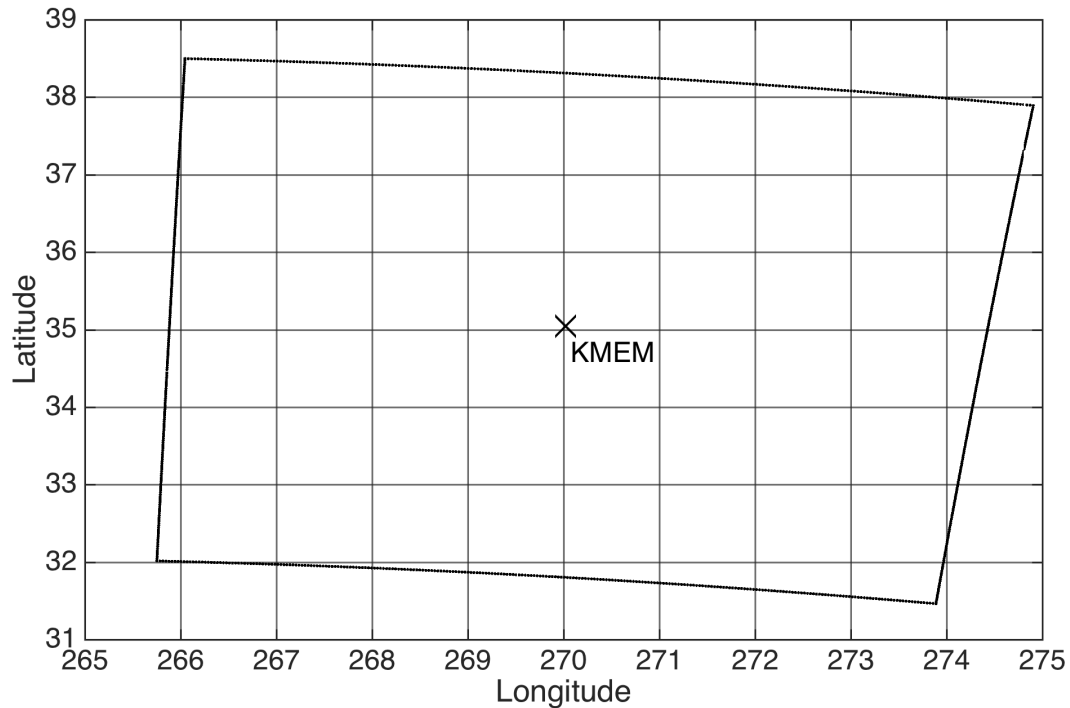


Figure 3. KMEM Domain.

the output to NetCDF format.

The GRIB2 default numbering convention for pressure levels start at the top of atmosphere down to the surface. A subroutine called `reversez.m` reorders each 3-D pressure level with the opposite direction. No changes in the other two dimensions (Latitude and Longitude) are performed. At this stage, the `wgrib2` processing of GRIB2 files is complete.

All temporary GRIB2 and NetCDF files are deleted. The output data files are all moved to the defined output directory and are in the Matlab “.mat” format. This concludes the description of the `extract.m` script. All subsequent processing is performed using other Matlab scripts. Note that the GRIB2 units are all SI, as are all subsequent output weather data fields.

## 7 Modifications to Weather Data Fields

Once the state variable weather fields in the GRIB2 files are extracted into separate 3-D files, certain modifications and additional calculations need to be performed. These include adjustments for altitude, wind velocity, and wind direction.

### 7.1 Altitude Modification

The Matlab script `hgtcorr.m` performs the altitude modification. Appendix C lists the `hgtcorr.m` script to convert geopotential pressure height to an altitude above mean sea level. This is performed by adding the location specific land elevation,

`msl.m`, to the `HGT` variables. The result remains in units of meters. The output file has a “Z” appended to the file name.

## 7.2 Vertical Wind Modification

Appendix D lists the `vvelcorr.m` script that converts the units of the vertical velocity variable as output from the GRIB2 files. Vertical velocity, `VVEL`, has units of Pa/s and must be converted to vertical wind velocity, `VVMS`, in units of m/s using the equation:

$$VVMS = (R_S * VVEL * T) / (g_Z * P) \quad (1)$$

where  $R_S$  is the specific gas constant for dry air,  $T_Z$  is temperature,  $g_Z$  is gravity, and  $P$  is pressure. The equation is applied at all altitudes. The 3-D output file has “VVMS” at the end of the file name.

## 7.3 Horizontal Wind Modification

Appendix E is a listing of the script `windsrot2d` that modifies the Lambert projection to true North-based winds. The HRRR model north direction is only parallel to true north at the center of the Lambert conic projection, thus the wind components must be rotated based on their domain location. This modification takes as input the `UGRD` and `VGRD` fields. The output files have either “UR” or “VR” appended for the modified U-component or V-component of winds, respectively.

The algorithm for horizontal wind modification is derived from information on the Rapid Update Cycle (RUC) Frequently Asked Questions website <http://ruc.noaa.gov/RUC.faq.html> [5]. On the website, an algorithm for the wind modification is implemented in FORTRAN code.

# 8 Derived Weather Fields

## 8.1 Radar Reflectivity

The script to generate radar reflectivity fields, `genrrrf.m` is listed in Appendix F. This script implements the equations for estimating radar reflectivity from state variables as derived in [6]. The output weather data field is designated with `RRF`, and has units dbZ.

This script implements Equations 2 - 6 from Daniels, et. al. (2013) [1], where radar reflectivity,  $Z$ , is the sum of reflectivities for rain, snow, and graupel.

## 8.2 Turbulence Parameter

HRRR model grid is too coarse for a turbulence field. The length scales of turbulence that impact aircraft dynamics are much shorter, typically on the order of 100 to 500 m. An alternative approach is to scale the intensity of a stochastic process (such as a Dryden Turbulence Model) with a measure of HRRR convection.

Appendix G is a listing of `tke.m`, a script that computes a turbulence parameter specifically for use in the RFD. As noted in [2], a turbulence parameter,  $T_p$ , was used to scale the random values from the Dryden Turbulence model [7]. This parameter is computed from the relation  $T_p = \gamma_1 Z_R + \gamma_2$ , where  $Z_R$  is radar reflectivity and scale factor  $\gamma_1$  and amplitude adjustment  $\gamma_2$  were determined empirically to produce the desired CMF/RFD ride quality. This equation yields turbulence linearly proportional with radar reflectivity. While turbulence is often experienced far from convection, it also occurs within convective regions.

The aerodynamic model has a turbulence transition height of 2500 ft AGL. When the aircraft model descends below transition altitude, the turbulence motion will start to fade out until reaching 50 ft. The CMF/RFD motion base actuators do not produce turbulence for aircraft altitudes below 50 ft.

### 8.3 Aircraft Icing

Supercooled liquid water (SLW) is assumed to be an indicator of aircraft icing. The Matlab script, `genslw.m`, listed in Appendix H, computes this weather data field. SLW mixing ratio has units kg/kg and are estimated using the following relation:

$$SLW = \begin{cases} CLWMR + RWMR & \text{for } 273.15 - 20 < T < 273.15 \\ 0 & \text{for } T > 273.15 \\ 0 & \text{for } T < 273.15 - 20 \end{cases} \quad (2)$$

## 9 Script Summary

All the Matlab scripts that are used to generate relevant weather data fields are listed in Table 2 along with input and output files. All input and output files have an “.mat” extension, with the exception of the GRIB2 files. These scripts are expected to be run in the order listed in the Table 2. Running `extract.m` creates the output file `memmetax.mat` that is used as input in all subsequent scripts. After all processing is complete, the new weather data fields (one per file) listed in Table 3, with the associated units, will exist in the user specified output directory.

## 10 Conclusions

This report documents a process and describes necessary code to implement a set of algorithms that create a 3-D weather data set. A referencable basis for the implemented equations is provided. This data set consists of weather data fields that can be used for simulation purposes. This report also provides usage details for the user to create customized weather data fields. An extension of this process includes the creation of 4-D weather in the simulation environment. Using the outlined process, four different weather data sets were created for use in the CMF/RFD.

Table 2. Matlab Script Summary		
Script Name	Input Files	Output Files
extract.m	GRIB2	HGT, TMP, CICE, SPFH VVEL, UGRD, VGRD, ABSV, CLWMR, RH, RWMR, SNMR, GRLE, memmetax
hgtcorr.m	HGT	Z
vvelcorr.m	Z, VVEL, TMP	VVMS
windsrot2d.m	UGRD, VGRD	UR, VR, WMAG, WDIR
genrrf.m	TMP, SPFH, GRLE, RWMR, CLWMR	RRF
tke.m	UR, VR, VVMS	TKE
genslw.m	TMP, CLWMR, RWMR, VVMS	SLW

Table 3. Resulting Weather Data Fields		
Field Name	Description	Units
Z	Geometric Height AGL	m
VVMS	Vertical Velocity	m/s
UR	Corrected U-component of Wind	m/s
VR	Corrected V-component of Wind	m/s
WMAG	Wind Magnitude	m/s
WDIR	Wind Direction	degrees
RRF	Radar Reflectivity	dBZ
TKE	Turbulence Parameter	%
SLW	Supercooled Liquid Water	kg/kg

## References

1. Daniels, T. S., Schaffner, P. R., Evans, E. T., Neece, R. T., & Young, S. D. (2012). Creating a realistic weather environment for motion-based piloted flight simulation. 2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC). pp. 1-22 doi:10.1109/dasc.2012.6382299 <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6382299&isnumber=6382265>
2. Daniels, T. S., Schaffner, P. R., Evans, E. T., and Young, S. D. (2013) Motion-based piloted flight simulation test results for a realistic weather environment. 2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC), pp. 5A3-1-5A3-15. doi: 10.1109/DASC.2013.6712595, <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6712595&isnumber=6712497>
3. Benjamin, S. G., Smirnova, T. G., Weygandt, S. S., Hu, M., Sahm, S. R., Jamison, B. D., Wolfson, M. M., Pinto, J. O. (2009). The HRRR 3-km storm-resolving, radar-initialized, hourly updated forecasts for air traffic management. The 89th American Meteorological Society Annual Meeting <https://ams.confex.com/ams/89AM/abstract.html>



[//ams.confex.com/ams/89annual/techprogram/paper\\_150430.htm](https://ams.confex.com/ams/89annual/techprogram/paper_150430.htm) (Last accessed April 8, 2016)

4. Alexander, C. R., Weygandt, S. S., Smirnova, T. G., Benjamin, S. G., Hoffmann, P., James, E. P., Koch, D. A. (2010) High resolution rapid refresh (HRRR): Recent enhancements and evaluation during the 2010 convective season. 25th Conference on Severe Local Storms. [https://ams.confex.com/ams/25SLS/techprogram/paper\\_175722.htm](https://ams.confex.com/ams/25SLS/techprogram/paper_175722.htm) (Last accessed April 8, 2016)
5. Benjamin, S. G. (2012). RUC Frequently Asked Questions. <http://ruc.noaa.gov/RUC.faq.html> (last accessed April 4, 2016)
6. Stoelinga, M. (2005) Description of the Simulated Equivalent Reflectivity Factor Calculation in RIP. [http://www.atmos.washington.edu/~stoeling/RIP\\_sim\\_ref.pdf](http://www.atmos.washington.edu/~stoeling/RIP_sim_ref.pdf) (Last accessed April 7, 2016)
7. MIL-STD-1797A Flying Qualities of Piloted Aircraft. (1997) Department of Defense [https://engineering.purdue.edu/~andrisan/Courses/AAE490F\\_S2008/Buffer/mst1797.pdf](https://engineering.purdue.edu/~andrisan/Courses/AAE490F_S2008/Buffer/mst1797.pdf) (Last accessed April 8, 2016)

## Appendix A

### runhrrr.m listing

```
1 % run_hrrr.m
% wrapper for wgrib2 to extract variable from a grib2 file

% grib2 files at:
%ftp.ncep.noaa.gov/pub/data/nccf/nonoperational/com/hrrr/prod/
6
% run hgtcorr.m to create Z files , add msl
% and vvelcorr.m to convert from pressure tendency to velocity
% use winds2d.m to compute wind mag and dir
% then windsrot2d.m to correct winds from Lambert to true north
11 % and genrrf.m to compute radar reflectivity
% tke.m generates the turbulence fields
% genslw.m creates the SLW field (aircraft icing)

setenv( 'DYLD_LIBRARY_PATH' , '/usr/local/bin/' );
16
% Windows version:
%gribdir='C:\Users\tsdaniel\Documents\VSST\HRRR\';
%gribdir='/Users/tsdaniel/Documents/HRRR/March2016MCC1/20160308/';

21 % User settings:

% Linux or Mac version:
gribdir='/Users/tsdaniel/Documents/HRRR/March2016MCC2/20160324/';
hrrrdir='/Users/tsdaniel/Documents/HRRR/';
26 % need target airport lat long
% memphis is 270,35, given as strings with spaces and colon
targetlat='_32:38_';
targetlong='_266:274_';

31 % end of user settings

NG=length( gribdir );

datestr=gribdir (NG-4:NG-1);
36
outdir=[gribdir , 'mat/'];
% create this dir
textout=['mkdir_',outdir];
if exist(outdir , 'dir') ~= 7
```

```

41|     system(textout);
    else
        %      error('%s directory already exists ',outdir)
        disp('Overwriting_existing_directory_contents. ');
    end
46| ext='*.grib2';
    gribfiles=strcat(gribdir,ext);
    fd=dir(gribfiles);
    NF=length(fd);
51| % HRRR data fields have 40 pressure levels, only need first
    % 36 counting from the ground up to get to ~40Kft, also given
    % as string with spaces
56| % don't change NP without changing options2 and options4 in extract.m
    NP=36;

    % executable string
    gbx='wgrib2_';
61|

    disp('HRRR_processing_started')

    extract
66| hgtcorr

    vvelcorr

71| windsrot2d

    genrrf

    tke
76| genslw

    disp('HRRR_processing_complete')

```

## Appendix B

### extract.m listing

```
1 % extract.m
  %
  % variables to extract from grib2 files
  allvars={'hgt'; 'tmp'; 'grle'; 'spfh'; 'vvel'; 'ugrd'; 'vgrd'; ...
6    'absv'; 'rwmr'; 'clwmr'};    % 'dpt'; 'cice'; 'snmr'; 'rh';};
  matchvar=upper(allvars);

  % process one var in first file to get lat lon data, then do all ...
  % check if memmetax is there, else make it
11 if exist ([hrrrdir, 'memmetax.mat'], 'file') ~=2

    buffer1='templ.grib2';
    buffer2='temp2.grib2';
    buffer3='temp3.nc';
16
    fname=fd(1).name;
    filename=strcat(gribdir, fname);

    % variable name is part of file name
21    varname=allvars(1);
    onm=strcat(datestr, fname(1:4), '_', fname(6:9), ...
        '_', char(varname));

    options1=['_match_':HGT:surface:_grib_out_', buffer1];
26
    options2=['_set_grib_type_same_small_grib', ...
        targetlong, targetlat, buffer2];

    options3=['_nc_nlev_36_netcdf_', buffer3];
31
    command1=[gbx filename options1];
    command2=[gbx buffer1 options2];
    command3=[gbx buffer2 options3];

36    [status1, varlist1] = system(command1);
    [status2, varlist2] = system(command2);
    [status3, varlist3] = system(command3);

    % clean up
```

```

41  delete(buffer1);
    delete(buffer2);

    % % output lat long plevels and other files once
    % read from netcdf file: lat long msl
46  latitude=ncread(buffer3,'latitude');
    longitude=ncread(buffer3,'longitude');
    msl_m=ncread(buffer3,'HGT_surface');

    delete(buffer3)

51  % now get pressure levels

    buffer1='temp1.grib2';
    buffer2='temp2.grib2';
56  buffer3='temp3.grib2';
    buffer4='temp4.nc';

    options1=['_match_:HGT:_grib_out_',buffer1];

61  % these are specific pressure levels in the wrfprs file
    options2=['_for_5:41_grib_out_',buffer2];
    % creates a netcdf file in 4-D {time;level;lat;long}

    options3=['_set_grib_type_same_small_grib',...
66  targetlong,targetlat,buffer3];

    options4=['_nc_nlev_36_netcdf_',buffer4];

    command1=[gbx filename options1];
71  command2=[gbx buffer1 options2];
    command3=[gbx buffer2 options3];
    command4=[gbx buffer3 options4];

    [status1, varlist1] = system(command1);
76  [status2, varlist2] = system(command2);
    [status3, varlist3] = system(command3);
    [status4, varlist4] = system(command4);

    % clean up
81  delete(buffer1);
    delete(buffer2);
    delete(buffer3)

    plevel=ncread(buffer4,'plevel');
86  % flip top bottom

```

```

    plevel=plevel(end:-1:1);
    %
    save memmetax latitude longitude plevel msl_m
    delete(buffer4);
91
else
    load([hrrrdir,'memmetax.mat']);
end
96 % now use lat lon data to get geo correct specific wx fields

for ii=1:NF % for each of the different events/times

    buffer1='templ.grib2';
101 buffer2='temp2.grib2';
    buffer3='temp3.grib2';
    buffer4='temp4.nc';

    fname=fd(ii).name;
106 filename=strcat(gribdir,fname);

    % for each of the vars at each event/time
    for jj=1:length(allvars)

111         % variable name is part of file name
        varname=allvars(jj);
        onm=strcat(datestr,fname(1:4),'_',fname(6:9),...
            '_','upper'(char(varname)));

116        currvar=char(matchvar(jj));
        vb1= sprintf('':%s:','',currvar);

        options1=['_match_', vb1, '_grib_out_',buffer1];

121        % these are specific pressure levels in the wrfprs file
        options2=['_for_5:41_grib_out_',buffer2];
        % creates a netcdf file in 4-D {time;level;lat;long}

        options3=['_set_grib_type_same_small_grib',...
126         targetlong,targetlat, buffer3];

        options4=['_nc_nlev_36_netcdf_', buffer4];

        command1=[gbx filename options1];
131        command2=[gbx buffer1 options2];
        command3=[gbx buffer2 options3];

```

```

command4=[gbx buffer3 options4];

[status1 , varlist1] = system(command1);
136 [status2 , varlist2] = system(command2);
[status3 , varlist3] = system(command3);
[status4 , varlist4] = system(command4);

varg=ncread(buffer4 , currvar);
141

% flip this 3-D variable top bottom
outvar=reversez(varg);

outstr=[currvar , '└─┘outvar;'];
146 eval(outstr);

savestr=['save└─┘',onm, '└─┘', currvar];
eval(savestr)

151 movename=[onm, '.mat'];
movefile(movename,outdir);

% clean up
delete(buffer1);
156 delete(buffer2);
delete(buffer3);
delete(buffer4);

end % of var loop
161 end % of event loop

```

## B.1 Subroutine reversez.m listing

```

function a=reversez(b)
2 %
%   reverses the page order of a 3D matrix
%
%   tsd    5/15/2012
%
7   [~,~,p]=size(b);

   a=b(:, :, p:-1:1);

return

```

## Appendix C

### hgtcorr.m listing

```
% hgtcorr.m

% Converts pressure altitude to geometric altitude
4
% size variables used in all subsequent processing scripts
[nlong, nlat]=size(latitude); % determine size
npl=length(plevel);

9 % create list of all HGT mat files in mat dir
cdm=dir(strcat(outdir, '*hgt.mat'));

for ii=1:NF % start loop

14     evalstr=['load_', outdir, cdm(ii,1).name]; % load data
    eval(evalstr);

    Z=zeros(nlong, nlat, npl); % allocate new matrix

19 % make new file name
    filename=strcat(cdm(ii,1).name(1:end-7), 'Z.mat');

    for jj=1:npl % add elevations
        Z(:, :, jj)=HGT(:, :, jj)+msl_m;
24    end

    % save as new output file
    evalstr=['save_', filename, '_Z'];
    eval(evalstr)
29

    movefile(filename, outdir)

end % end of N loop
```



## Appendix D

### vvelcorr.m listing

```
% vvelcorr.m
% vertical velocity correction
3 % data from HRRR has pressure tendency
%
% Run this after extract and after hgtcorr
% uses temperature, vvel, Z

8 % Converts pressure tendency (Pa/s)
%   to spatial velocity (m/s)
%   use version of hydrostatic eqn
%    $W = -w R_s T / P gZ$ 
%
13 % need gravity at altitude
%    $gZ = g0 (R_e / (R_e + Z))^2$ 

% some constants
Re=6375000; % effective Earth radius in m
18 g0=9.80665; % gravity at surface
Rs=287.04; % specific gas constant for dry air

% convert pressure to Pa
plevel_pa=plevel*100;
23
% create list of Z mat files in mat dir
zfiles=dir(strcat(outdir, '*Z.mat'));

% create list of vvel mat files in mat dir
28 vfiles=dir(strcat(outdir, '*vvel.mat'));

% create list of tmp mat files in mat dir
tfiles=dir(strcat(outdir, '*tmp.mat'));

33 for ii=1:NF % start loop

    % load Z data
    evalstr=['load_', outdir, zfiles(ii,1).name];
    eval(evalstr);

38
    % load V data
    evalstr=['load_', outdir, vfiles(ii,1).name];
```

```

    eval(evalstr);

43  % load temperature data
    evalstr=['load_',outdir,tfiles(ii,1).name];
    eval(evalstr);

    VVMS=zeros(nlong,nlat,npl); % allocate space

48  % compute gravity (Z)
    gZ= g0 .* ( Re ./ (Re + Z )).^2;

    % make new file name
53  filename=strcat(zfiles(ii,1).name(1:end-5),'VVMS.mat');

    % compute spatial velocity
    for jj=1:nlong
        for kk=1:nlat
58          for pp=1:npl
              VVMS(jj,kk,pp)=Rs.*VVEL(jj,kk,pp).*TMP(jj,kk,pp)...
                  ./ (gZ(jj,kk,pp) .* plevel_pa(pp));
          end
        end
63    end

    % save as new output file
    evalstr=['save_',filename,'_VVMS'];
    eval(evalstr)

68  movefile(filename,outdir)

end % end of N loop

```

## Appendix E

### windsrot2d.m listing

```
% winds2d.m
% compute horizontal wind mag and dir

4 % From Stan Benjamin, http://ruc.noaa.gov/RUC.faq.html
%
% FORTRAN code to implement wind correction for RUC data
%
%      PARAMETER ( ROTCON_P = 0.422618 )
9 %      PARAMETER ( LON_XX_P = -95.0 )
% C** ROTCON_P R WIND ROTATION CONSTANT, = 1 FOR POLAR STEREO
% C**              AND SIN(LAT_TAN_P) FOR LAMBERT CONFORMAL
% C** LON_XX_P R MERIDIAN ALIGNED WITH CARTESIAN X-AXIS(DEG)
% C** LAT_TAN_P R LATITUDE AT LAMBERT CONFORMAL PROJECTION
14 % C**              IS TRUE (DEG)
%      PARAMETER ( LAT_TAN_P = 25.0 )
%
%      do j=1,ny-p
%      do i=1,nx-p
19 %          angle2 = rotcon_p*(olon(i,j)-lon_xx_p)*0.017453
%          sinx2 = sin(angle2)
%          cosx2 = cos(angle2)
%          do k=1,nzp-p
%              ut = u(i,j,k)
24 %              vt = v(i,j,k)
%              un(i,j,k) = cosx2*ut+sinx2*vt
%              vn(i,j,k) = -sinx2*ut+cosx2*vt
%          end if
%      end do
29 % end do

% implmented here for HRRR instead of RUC...

% create list of all mat files in mat dir
34 ufiles=dir(strcat(outdir,'*ugrd.mat'));
vfiles=dir(strcat(outdir,'*vgrd.mat'));

% start loop
for ii=1:NF
39     % load data
```

```

evalstr=['load_', outdir, ufiles(ii,1).name];
eval(evalstr);
evalstr=['load_', outdir, vfiles(ii,1).name];
44 eval(evalstr);

% make new file name
filename1=strcat(ufiles(ii,1).name(1:end-8), 'WMAG.mat');
filename2=strcat(ufiles(ii,1).name(1:end-8), 'WDIR.mat');
49 filename3=strcat(ufiles(ii,1).name(1:end-8), 'UR.mat');
filename4=strcat(ufiles(ii,1).name(1:end-8), 'VR.mat');

WMAG = sqrt(UGRD.^2 + VGRD.^2);
WDIR = atan2d(VGRD, UGRD);
54

% these two variables are defined from GRIB outputs:
% ie. wgrib2 -grid
% Lat{D,1,2} = latitude where the Lambert conformal
% cone is tangent to the surface of the earth
59 LatD = 38.5;
%elonv = LoV = longitude of the center of the projection
LoV = 262.5;

% elon1 - longitude of lower left point of grid (1,1)
64 % east longitude used throughout;
elon=237.28;

deg_to_radians = 0.0174533;

69 % theta is degrees to rotate winds from RUC coords

% rotate direction appropriately for the RUC grid */
% this means that S winds near the E coast need to pick up
% an W component wrt true N. So rotate the wind vectors
74 % clockwise (+) near the E coast.

rotcon = sin(LatD*deg_to_radians);
dlon = LoV - elon;
while(dlon > 180)
79     dlon = dlon - 360;
end
while (dlon < -180)
    dlon = dlon + 360;
end
84 theta = - rotcon*(dlon);
wdnu=WDIR*57.2958 + theta + 180. + 0.5;
if(wdnu > 360)

```

```

      wdu = wdu- 360;
    end
89   if(wdu < 0)
      wdu = wdu +360;
    end

    WDIR=wdu;
94
    UR=WAG.*cos(WDIR);
    VR=WAG.*sin(WDIR);

    % save as new output files
99   evalstr=['save_',filename1,'_WAG'];
    eval(evalstr)
    evalstr=['save_',filename2,'_WDIR'];
    eval(evalstr)
    evalstr=['save_',filename3,'_UR'];
104   eval(evalstr)
    evalstr=['save_',filename4,'_VR'];
    eval(evalstr)

    movefile(filename1,outdir)
109   movefile(filename2,outdir)
    movefile(filename3,outdir)
    movefile(filename4,outdir)

end    % end of N loop

```

## Appendix F

### genrrf.m listing

```
% genrrf.m
2 % compute radar reflectivity in dBZ
%
% from RIP4 code for dbzcalc.f,
% Mark Stoelinga (stoeling@atmos.washington.edu)

7 % switches for computing...
in0s=0;
in0g=0;
in0r=0;

12 % assume frozen particles that are at a temperature above
% freezing are assumed to scatter as a liquid particle
iliqskin=1; % 0 is off, 1 is on

celkel=273.15; % need to check this

17 % Constant intercepts
Rdry=287; % dry air gas constant J/K/kg
rn0_r = 8.e6; % m-4
rn0_s = 2.e7; % m-4
22 rn0_g = 4.e6 ; % m-4

% Constants used to calculate variable intercepts
r1=1.e-15;
ron=8.e6;
27 ron2=1.e10;
son=2.e7;
gon=5.e7;
ron_min = 8.e6;
ron_qr0 = 0.00010;
32 ron_delqr0 = 0.25*ron_qr0;
ron_const1r = (ron2-ron_min)*0.5;
ron_const2r = (ron2+ron_min)*0.5;
gamma_seven = 720.;
rhowat = 1000; % water density kg m-3
37 rho_r = 1000.; % 1000. kg m-3
rho_s = 100.; % kg m-3
rho_g = 400.; % kg m-3
alpha = 0.224;
```

```

factor_r = gamma_seven * 1.e18 * (1./(pi*rho_r))^1.75;
42 factor_s = gamma_seven * 1.e18 * (1./(pi*rho_s))^1.75 ...
    * (rho_s/rhowat)^2 * alpha;
factor_g = gamma_seven * 1.e18 * (1./(pi*rho_g))^1.75 ...
    * (rho_g/rhowat)^2 * alpha;

47 factorb_s=factor_s;
factorb_g=factor_g;

% create pressure field
PR=zeros(nlong, nlat, npl); % pressure aka prs
52 for ii=1:npl
    PR(:, :, ii)=repmat(plevel_pa(ii),[nlong nlat]);
end

% create list of all needed mat files in mat dir
57 tfiles=dir(strcat(outdir, '*tmp.mat'));
qfiles=dir(strcat(outdir, '*spfh.mat'));
cfiles=dir(strcat(outdir, '*clwmr.mat'));
rfiles=dir(strcat(outdir, '*rwmr.mat'));
gfiles=dir(strcat(outdir, '*grle.mat'));
62 for ii=1:NF % start loop, first load data

    % TMP aka tmk
    evalstr=['load_', outdir, tfiles(ii,1).name];
67 eval(evalstr);
    % SPFH aka qvp
    evalstr=['load_', outdir, qfiles(ii,1).name];
    eval(evalstr);
    % CLWMMR aka qsn
72 evalstr=['load_', outdir, cfiles(ii,1).name];
    eval(evalstr);
    % RWMMR aka qra
    evalstr=['load_', outdir, rfiles(ii,1).name];
    eval(evalstr);
77 % GRLE aka qgr
    evalstr=['load_', outdir, gfiles(ii,1).name];
    eval(evalstr);

    % compute virtual temp for density
82 VT = virtual(TMP,SPFH);

    % compute density
    rhoair = PR ./ (Rdry .* VT); % air density

```

```

87  % fine where ice/snow/graupel is becoming water
    lqdex=find(TMP > celkel);

    % make new file name
    filename=strcat( tfiles(ii,1).name(1:end-8),'_RRF.mat');
92
    %      compute rrf
    %      Calculate variable intercept parameters if wanted
    if (in0s==1) % N_0s as in Thompson et al.
        temp_c = amin1(-0.001, TMP-celkel);
97        sonv = amin1(2.0e8, 2.0e6*exp(-0.12*temp_c));
    else
        sonv = rn0_s;
    end

102    if (in0g==1) % N_0g as in Thompson et al.
        gonv = gon;
        if (GRLE>r1)
            gonv = 2.38.*(pi.*rho_g./(rhoair.*GRLE)).^0.92;
            gonv = max(1.e4, min(gonv,gon));
107        end
    else
        gonv = rn0_g;
    end

112    if (in0r==1) % N_0r as in Thompson et al.
        ronv = ron2;
        if (RWMR>r1)
            ronv = ron_const1r.*tanh((ron_qr0-RWMR)./ron_delqr0)...
            + ron_const2r;
117        end
    else
        ronv = rn0_r;
    end

122    % Total equivalent reflectivity factor (z_e, in mm^6 m^-3) is
    % the sum of z_e for each hydrometeor species:
    z_e = factor_r .* (rhoair.*RWMR).^1.75 ./ ronv^.75 ...
% rain
        + factorb_s .* (rhoair.*CLWMR).^1.75 / sonv^.75 ...
% snow
        + factorb_g .* (rhoair.*GRLE).^1.75 / gonv^.75; % graupel
127
    % correct for melting ice/snow/graupel
    if (iliqskin==1)
        factorb_s=factor_s/alpha;

```



```

132     factorb_g=factor_g/alpha;
    z_e(lqdex) = ...
        factor_r .* (rhoair(lqdex).*RWMR(lqdex)).^1.75...
        ./ ronv^.75 ... % rain
    + factorb_s .* (rhoair(lqdex).*CLWMR(lqdex)).^1.75...
    / sonv^.75 ... % snow
137 + factorb_g .* (rhoair(lqdex).*GRLE(lqdex)).^1.75...
    / gonv^.75; % graupel
    end

    % Adjust small values of Z-e so that dBZ is no lower than -20
142 z_e = max(z_e,.01);

    % Convert to dBZ
    RRF = 10. * log10(z_e);

147 % save as new output file
    evalstr=['save_',filename,'_RRF'];
    eval(evalstr)

    movefile(filename,outdir)
152 end % end of N loop

```

## Appendix G

### tke.m listing

```
% tke.m
2 % compute tke per mass from winds
%

% create list of all mat files in mat dir
ufiles=dir(strcat(outdir, '*ur.mat'));
7 vfiles=dir(strcat(outdir, '*vr.mat'));
wfiles=dir(strcat(outdir, '*vvms.mat'));

% create averaging filter
h=ones(3);
12

% start loop
for ii=1:NF

    % load data
17    evalstr=['load_', outdir, ufiles(ii,1).name];
    eval(evalstr);
    evalstr=['load_', outdir, vfiles(ii,1).name];
    eval(evalstr);
    evalstr=['load_', outdir, wfiles(ii,1).name];
22    eval(evalstr);

    % allocate
    URDM=zeros(nlong, nlat, npl);
    VRDM=zeros(nlong, nlat, npl);
27    VVMSM=zeros(nlong, nlat, npl);

    % make new file name
    filename1=strcat(ufiles(ii,1).name(1:end-7), '_TKE.mat');

32    % compute local means
    for ik=1:npl %for each level
        URDM(:, :, ik) = filter2(h, UR(:, :, ik));
        VRDM(:, :, ik) = filter2(h, VR(:, :, ik));
        VVMSM(:, :, ik) = filter2(h, VVMS(:, :, ik));
37    end

    % compute gusts by removing mean
    URDP=UR-URDM;
```

```

VRDP=VR-VRDM;
42  VVMSP=VVMS-VVMSM;

    % compute tke from gusts
    TKE=(URDP.^2 + VRDP.^2 + VVMSP.^2)/2;

47  % save as new output file
    evalstr=['save_',filename1,'_TKE'];
    eval(evalstr)

    %      movefile(filename1,matdir)
52  movefile(filename1,outdir)

end    % end of N loop

```

## Appendix H

### genslw.m listing

```
1 % genslw.m
  % compute super cooled liquid water
  %

  celkel=273.15;

6  % create pressure field
  PR=zeros(nlong, nlat, npl); % pressure aka prs
  for ii=1:npl
    PR(:, :, ii)=repmat(plevel_pa(ii), [nlong nlat]);
11 end

  % create list of all needed mat files in mat dir
  tfiles=dir(strcat(outdir, '*tmp.mat'));
  %qfiles=dir(strcat(outdir, '*spf.h.mat'));
16 cfiles=dir(strcat(outdir, '*clwmmr.mat'));
  rfiles=dir(strcat(outdir, '*rwmr.mat'));
  %ifiles=dir(strcat(outdir, '*cice.mat'));
  vfiles=dir(strcat(outdir, '*vms.mat'));

21 for ii=1:NF % start loop

    % load data
    % TMP aka tmk
    evalstr=['load_', outdir, tfiles(ii,1).name];
26 eval(evalstr);
    % SPFH aka qvp
    %evalstr=['load_', outdir, qfiles(ii,1).name];
    %eval(evalstr);
    % CLWMMR aka qsn
31 evalstr=['load_', outdir, cfiles(ii,1).name];
    eval(evalstr);
    % RWMR aka gra
    evalstr=['load_', outdir, rfiles(ii,1).name];
    eval(evalstr);
36 % CICE
    %evalstr=['load_', outdir, ifiles(ii,1).name];
    %eval(evalstr);
    % VMS
    evalstr=['load_', outdir, vfiles(ii,1).name];
```

```

41      eval(evalstr);

      % make new file name
      filename=strcat( tfiles(ii,1).name(1:end-8), '_SLW.mat' );

46      [nr,nc,np]=size(TMP);
      SLW=zeros(nr,nc,np);
      buffer=ones(nr,nc,np);

      % compute slw from all sources of water and some ice
51      %   SLW = CLWMR + RWMR + SPFH + 0.5 * CICE;
      SLW = CLWMR + RWMR;
      SLW(TMP > celkel) = 0;      % too warm
      SLW(TMP < celkel-25) = 0;   % too cold
      % buffer(VVMS < 0) = 0.5;   % downwind scavenged by graupel
56      %SLW=SLW .* buffer;      % but only by half
      %SLW(SLW < 0) = 0;        % error check

      % save as new output file
      evalstr=['save_',filename, '_SLW'];
61      eval(evalstr)

      movefile(filename, outdir)

end    % end of N loop

```

REPORT DOCUMENTATION PAGE					Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p><b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>						
1. REPORT DATE (DD-MM-YYYY) 01-07-2016		2. REPORT TYPE Technical Memorandum		3. DATES COVERED (From - To) 10/2012-4/2016		
4. TITLE AND SUBTITLE Using 3-D Numerical Weather Data in Piloted Simulations				5a. CONTRACT NUMBER		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Daniels, Taumi, S.				5d. PROJECT NUMBER		
				5e. TASK NUMBER		
				5f. WORK UNIT NUMBER 330693.04.80.07.07		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, Virginia 23681-2199				8. PERFORMING ORGANIZATION REPORT NUMBER L-20718		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001				10. SPONSOR/MONITOR'S ACRONYM(S) NASA		
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) NASA-TM-2016-219216		
12. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category 61 Availability: NASA STI Program (757) 864-9658						
13. SUPPLEMENTARY NOTES An electronic version can be found at <a href="http://ntrs.nasa.gov">http://ntrs.nasa.gov</a> .						
14. ABSTRACT This report describes the process of acquiring and using 3-D numerical model weather data sets in NASA Langley's Research Flight Deck (RFD). A set of software tools implement the process and can be used for other purposes as well. Given time and location information of a weather phenomenon of interest, the user can download associated numerical weather model data. These data are created by the National Oceanic and Atmospheric Administration (NOAA) High Resolution Rapid Refresh (HRRR) model, and are then processed using a set of Mathworks' Matlab <sup>TM</sup> scripts to create the usable 3-D weather data sets. Each data set includes radar reflectivity, water vapor, component winds, temperature, supercooled liquid water, turbulence, pressure, altitude, land elevation, relative humidity, and water phases. An open-source data processing program, <b>wgrib2</b> , is available from NOAA online, and is used along with Matlab scripts. These scripts are described with sufficient detail to make future modifications. These software tools have been used to generate 3-D weather data for various RFD experiments.						
15. SUBJECT TERMS Numerical Modeling, Weather Simulation						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Information Desk ( <a href="mailto:help@sti.nasa.gov">help@sti.nasa.gov</a> )	
U	U	U	UU	39	19b. TELEPHONE NUMBER (Include area code) (757) 864-9658	



